IP2User – Identifying the username of an IP Address in Network-Related Events

Asaf Shabtai[1], Idan Morad[1], Eyal Kolman[2], Ereli Eran[2], Alex Vaystikh[2], Eyal Gruss[2], Lior Rokach[1], Yuval Elovici[1]
[1]Dept. of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel
[2]EMC/RSA, Israel
{shabtaia, idanmora}@bgu.ac.il, {eyal.kolman, ereli.eran, alex.vaystikh, eyal.gruss}@rsa.com, {liorrk, elovici}@bgu.ac.il

***Abstract*—network devices deployed in organizations (Firewall, IDS, routers, antivirus, servers, etc.) logs users' activity as events. Based on these events users' behavioral profiles can be derived in order to detect anomalies, indicating potential attacks. The identifier of a user in most cases is the user's organizational username. While events are always logged with the source IP address they are not always logged with the relevant username and therefore, many of the collected events are not directly linked with the appropriate user. In this paper we describe a method for associating an IP address with an actual username based on a set of logged events. This is crucial precondition for generating an accurate user's profile. The proposed method was evaluated using real large datasets (logs) and showed 88% accuracy in the identification of usernames.**

Keywords-security and event management, anomaly detection, user profiling

I. INTRODUCTION

Users in organizations regularly access various internal and external computational resources (application and files servers, mail servers, the Internet etc.) Access to these resources can be either from within the company's network or from an external network (via secured VPN connection).

Such user activity is logged as events by various devices (Firewalls, DLP systems, IDSs, routers, antivirus, servers, etc.) Each event is logged with relevant attributes such as timestamp, username, source and destination IP/port, type of the device that logged the event, protocol, bytes sent/received and more. These events are then collected by Security Information and Event Management (SIEM) systems for further processing and analysis in an attempt to detect cyber-attacks. For example, users' behavioral profiles can be derived based on the collected events in order to detect anomalies or malicious activity [1]. The identifier of a user in most cases is the user's organizational username.

While events are always logged with the source IP address, in most of the cases they are logged without the relevant username and thus cannot be linked with the relevant user. In order to profile users as accurate as possible, a pre-processing phase in which the events are linked with the relevant user is essential [2].

In this paper we describe a method for linking an IP address of a given event with its genuine username. Assigning the correct username to an IP address is challenging due to the fact that most of the IP addresses are assigned dynamically to computers once they are connected to the organizational network. The IP address is allocated for a certain period of time (usually several days) after which it is released and may be allocated to a different computer. In addition, some events are generated by servers on behalf of a specific computer (i.e., user). In these events, the IP address is the server IP, but the username in the event is the username of the user. Using such events in the linkage process may result in wrong assignment of usernames to IP addresses and therefore, we would like to identify and remove them.

II. THE PROPOSED LINKAGE METHOD

The general idea is to use events containing both the IP address and the username (i.e., *labeled events*) in order to create a lookup table. The lookup table is then used for assigning a username to the IP address of an event with missing username (i.e., *unlabeled event*). Fig. 1 illustrates the proposed linkage approach. The username of the unlabeled event (e_2) is set to be 'Alice' based on the nearest (preceding) labeled event with the same IP address (e_1). However, in order to better link between an IP address to the relevant username we need first to identify and filter out events with server IP addresses.

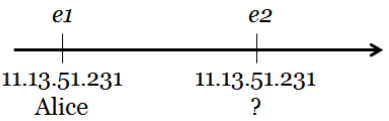

Figure 1. The username of the unlabeled event e_2 is set according to the nearest (preceding) labeled event with the same IP address (e_1).

A. Detecting Servers

In order to detect server IPs we rely on the fact that IPs of servers are usually static (i.e., change rarely if at all) and are expected to appear frequently in the logs with different usernames. Therefore, we define two parameters: a time window, *w*, and a threshold *s*. An IP address that is logged within the time window *w* with a number of different usernames that is greater than *s* is marked as a server.

B. Deriving the Lookup Table

The next step, after removing events with server IP addresses, is creating the lookup table that will be used for assigning usernames to unlabeled events. The lookup table is generated from the list of labeled events sorted by timestamp. Each IP address in the lookup table points to a sorted list of time intervals, each labeled with the username to which the IP address was allocated during that time interval and (if applicable) the device type that logged the event (see example in Fig. 2).

Figure 2. An example of the lookup table. The Source IP address 124.176.110.13 points to a list of sorted timestamp of labeled events; each points to the relevant username, end time and device type name.

The start time of an interval is set as the *timestamp of a labeled event*. The end time of the interval can be set according to one of two optional modes of operations. In the first mode, '*allow concurrent IPs*', we assume that a single username may be concurrently connected to more than one IP address (e.g., when the user connects with two different devices). In this mode, the end time of the interval is set only when there is another labeled event with the same IP address but with a different username (meaning that the IP address was assigned to a different username). In the second mode, '*no concurrent IPs*', the same username cannot be assigned at the same time to two different IPs and we expect to see the username alternating between the IP addresses. Thus, in this mode the end time of an interval is set as the earliest timestamp of the following events: (1) consecutive labeled event with the same IP address but with a different username (i.e., the IP address is assigned to a different username); and, (2) consecutive labeled event of the same username but with a different IP address (i.e., the user switches to another IP address). The first mode of operation (*allow concurrent IPs*) allows labeling of all non-server IP addresses. However, it increases the risk of incorrect labeling, especially in cases when the time gap between two labeled events with the same IP address but different usernames is large. The '*no concurrent IPs*' mode, in this sense is expected to be more accurate; however, it introduces cases in which unlabeled events cannot be linked with a username. An example of this case is presented in Fig. 3. It can be observed that from 10pm to 11pm no username is assigned to the IP address 11.13.51.231 (we termed such interval as '*unknown interval*'). Thus, unlabeled events falling in unknown intervals cannot be labeled. In order to increase the number of predicted events we apply the following heuristic. If an unlabeled event "falls" within an 'unknown interval', we check the nearest preceding and nearest consecutive intervals and if the usernames assigned to these two intervals are the identical, we assign the same username to the unlabeled event.

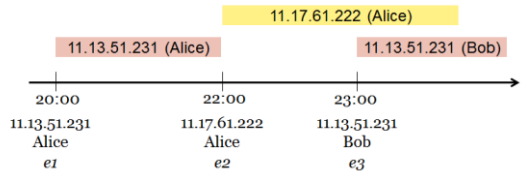

Figure 3. The username 'Alice' is assigned to IP address 11.13.51.231 from 8pm. The end time of the entry is 10pm when the user 'Alice' has changed its IP address to 11.17.61.222.

C. *Assigning username to unlabeled events*

The 'allow concurrent IPs' method can be applied for both real-time and retrospective username assignment. Given an unlabeled event *e* we look for the most recent time interval in the lookup table with the same IP address and the event *e* is assigned with the username of that interval. When a *labeled* event is logged, the lookup table can be updated accordingly. The 'no concurrent IPs' mode can be applied only for retrospective username assignment. This is because we are evaluating the nearest preceding and nearest consecutive intervals in the lookup table having the same IP of the unlabeled event *e*.

III. EVALUATION

In order to evaluate the proposed method we used two real-life datasets, each containing one week of data, collected from an operational network infrastructure. Note that 76% of the events are logged without usernames. For the evaluation we used only the labeled events in each dataset. The first dataset contains 19,547,156 labeled events and the second dataset contained 20,940,507 labeled events. The events were logged by 15 different network devices (firewalls, IDSs, DLP, mobile gateway etc.) In order to perform the evaluation we chose the events of one device (email and Web security device) as the test-set (i.e., we attempt to identify the usernames of events logged by this device) while the rest of the events were used for deriving the lookup table. The number of test events in the first dataset is 308,037 and in the second dataset 244,923. The results of the experiments are presenting in Table 1. As expected, it can be seen that the 'no concurrent IPs' mode achieved higher accuracy (approx. 1.3%) while leaving 2.8-5.6% of the events unlabeled.

TABLE I. EVALUATION RESULTS

	Dataset 1	Dataset 2
allow concurrent IPs	85.2% (0%)	87.0% (0%)
no concurrent IPs	86.7% (5.6%)	88.2% (2.76%)

IV. CONCLUSION AND FUTURE WORK

We propose a simple and accurate method for determining the usernames of IP address. We intend to improve the performance of the proposed method by incorporating machine learning algorithms that will predict the time in which an IP address was assigned to a different user based on the user's activity. In addition, we will attempt to identify static IPs assigned to users which, we believe, will help reducing the false positives.

ACKNOWLEDGMENTS

This research was conducted in cooperation with EMC/RSA, Israel.

REFERENCES

[1] C. You, *et al.*, "Specializing network analysis to detect anomalous insider actions," SecurityInformatics, 1.1, pp. 1-24, 2012.

[2] G. Zhang, and B. Reuther, "A model for user based traffic accounting," In 31st Conference on Software Engineering and Advanced Applications, pp. 354-361, 2005.